Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de recherche des recettes via le champ de recherche et/ou les tags des filtres	**Fonctionnalité 1**
Problématique : L'idée est de bénéficier d'une recherche plus rapide ainsi que la plus fluide possible grâce à la fonctionnalité de « recherche » à une recette correspondant au besoin de l'utilisateur via les ingrédients, les noms de recettes ou encore bien sa description.	

Option 1 : Algorithme via des boucles avancées (filter, forEach etc…).	
Avantages ⊕ Nombres de lignes de code inférieur à l'autre option, code plus structuré et lisible ⊕ Algorithme plus rapide ⊕ Permet une meilleure lisibilité de code	**Inconvénients** ⊖ prise de connaissances des méthodes
Nombre de champs minimum à remplir à l'inscription : 1 champ de recherche **Nombres de sélecteurs possibles :** 3 sélecteurs (les ingrédients, les appareils et les ustensiles). **Nombre de champs minimum à remplir à sélectionner:** 0 champ de recherche et 0 sélecteur permettent le renvoie de l'intégralité des recettes.	

Option 2 : Algorithme via des boucles natives (for..while etc…).	
Avantages ⊕ Savoir utiliser des connaissance de bases de JS afin de pouvoir élaborer son algorithme	**Inconvénients** ⊖ La compréhension du code peut être plus complexe. ⊖ Algorithme moins rapide
Nombre de champs minimum à remplir à l'inscription : 1 champ de recherche **Nombres de sélecteurs possibles :** 3 sélecteurs (les ingrédients, les appareils et les ustensiles). **Nombre de champs minimum à remplir à sélectionner:** 0 champ de recherche et 0 sélecteur permettent le renvoie de l'intégralité des recettes.	

Solution retenue :
L'option 1 est retenue puisqu'elle va permettre d'avoir une recherche plus pertinente et donc plus rapide pour nos utilisateurs. Côté développement, elle permet d'avoir une solution algorithmique beaucoup plus lisible et donc maintenable que l'option 2.

Version(foreach, filter, map, reduce)

```js
/**
 * Filtre les recettes en fonction du champ saisi par l'utilisateur dans la barre de recherche
 * @param {Object[]} recipesTofilter - Liste des recettes à filtrer
 * @param {string} inputValue - Valeur du champ dans la barre de recherche
 * @returns
 */
function filterRecipesWithGlobalInput(recipesTofilter, inputValue) {
  const inputValueFormatted = inputValue.trim().toLowerCase();

  const recipesListToDisplay = recipesTofilter.filter((recipe) => {
    let recipeIsMatching = false;
    // Vérifie si la valeur saisie correspond à un titre, à la description ou à un ingrédient de la carte
    if (
      recipe.name.toLowerCase().includes(inputValueFormatted) ||
      recipe.description.toLowerCase().includes(inputValueFormatted)
    ) {
      recipeIsMatching = true;
    }
    recipe.ingredients.forEach(({ ingredient }) => {
      const ingredientNameFormatted = ingredient.toLowerCase();
      if (ingredientNameFormatted.includes(inputValueFormatted)) {
        recipeIsMatching = true;
      }
    });
    return recipeIsMatching;
  });
  return recipesListToDisplay;
}
```

Version(while, for)

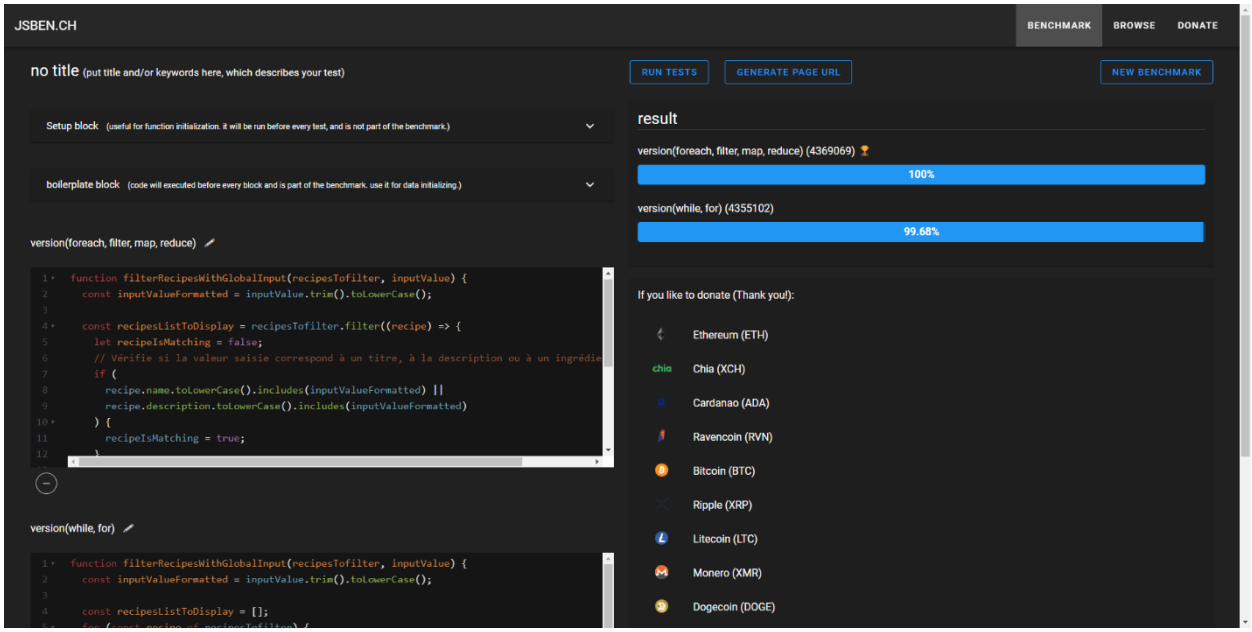
```js
/**
 * Filtre les recettes en fonction du champs tapé par l'utilisateur dans la barre de recherche
 * @param {Object[]} recipesTofilter - Liste des recettes à filtrer
 * @param {string} inputValue - Valeur du champ dans la barre de recherche
 * @returns
 */
function filterRecipesWithGlobalInput(recipesTofilter, inputValue) {
  const inputValueFormatted = inputValue.trim().toLowerCase();

  const recipesListToDisplay = [];
  for (const recipe of recipesTofilter) {
    let recipeIsMatching = false;
    // Vérifie si la valeur saisie correspond à un titre, à la description ou à un ingrédient de la carte
    if (
      recipe.name.toLowerCase().includes(inputValueFormatted) ||
      recipe.description.toLowerCase().includes(inputValueFormatted)
    ) {
      recipeIsMatching = true;
    }

    for (const { ingredient } of recipe.ingredients) {
      const ingredientNameFormatted = ingredient.toLowerCase();
      if (ingredientNameFormatted.includes(inputValueFormatted)) {
        recipeIsMatching = true;
      }
    }

    if (recipeIsMatching) {
      recipesListToDisplay.push(recipe);
    }
  }
  return recipesListToDisplay;
}
```

Test de performance des algorithmes:



La version(foreach, filter, map, reduce) est plus performante que la version version(while, for).